

UNCLASSIFIED

**Defense Technical Information Center
Compilation Part Notice**

ADP010972

TITLE: SoC: Une Nouvelle Approche de l'Amelioration des Performances des Systemes pour Combattre les Problemes de Disponibilite a Long Terme [SoC: A New Approach to Enhance System Performances and to Combat the Long-Term Availability Issue]

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: Strategies to Mitigate Obsolescence in Defense Systems Using Commercial Components [Strategies visant a attenuer l'obsolescence des systemes par l'emploi de composants du commerce]

To order the complete compilation report, use: ADA394911

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:

ADP010960 thru ADP010986

UNCLASSIFIED

SoC : Une nouvelle approche de l'amélioration des performances des systèmes pour combattre les problèmes de disponibilité à long terme

SoC : A New Approach to Enhance System Performances and to Combat the Long-Term Availability Issue

(janvier 2001)

Ph. Butel

Head of ASIC Design

Matra BAe Dynamics

20-22, rue Grange Dame Rose - BP150

78141 Vélizy Villacoublay Cedex

France

e-mail: pbute@matra-def.fr

RESUME

Les applications militaires ayant perdu leur leadership dans le domaine de l'électronique, elles auront de plus en plus à utiliser des technologies civiles. Il faudra apprendre à les utiliser ou à les adapter à nos spécificités, par exemple faibles volumes en production, température de fonctionnement élevée... L'utilisation de ce que l'on a pris l'habitude d'appeler « composants sur étagère » continuera même si l'assurance de pouvoir les approvisionner sur le long terme est un souci non négligeable.

Mais une autre technologie, également issue du Civil, paraît prometteuse : Les « System on Chip » ou « SoC ». En d'autres termes, la possibilité d'intégrer dans un seul circuit ou des circuits en nombre réduit un calculateur complet, répondant, par exemple, à une application de pilotage / guidage pour missile. Cette approche est maintenant bien établie dans le monde civil et industriel tel que les télécommunications, mais encore relativement peu implantée dans les systèmes de défense.

Il s'agit en fait d'une technologie ASIC (Application Specific Integrated Circuit), mais intégrant jusqu'à plusieurs millions de portes. Pour pouvoir maîtriser la complexité de la phase de conception en terme de coût et de délais, les SoC sont largement basés sur la notion de réutilisation de blocs fonctionnels : les « Intellectual Properties » ou « IP ». En fait, ces IP ne sont rien d'autres que des composants sur étagère mais virtuels, donc indépendants d'une quelconque technologie. Ils peuvent soit achetés soit être issus de conceptions précédentes. Les avantages sont nombreux, par exemple :

- Il est possible de concevoir le SoC sur la gamme de température voulue.
- En cas d'obsolescence d'une technologie, la société utilisatrice étant propriétaire de la définition du circuit peut migrer vers une technologie plus récente...

Certaines difficultés restent, bien entendu, à surmonter tel que, et de manière non exhaustive :

- L'accès aux fonderies, en cas de sélection d'une technologies ASIC (par opposition à des Programmable Logic Devices : PLD) du fait des faibles volumes ;
- La durée des plannings de développement ;
- Le coût des composants virtuels.

Tous ces points sont passés en revue dans ce document.

INTRODUCTION

L'évolution des marchés Militaires

L'utilisation de concepts civils pour des applications militaires est une tendance qui peut déjà être constatée et qui va sûrement s'amplifier. Une telle démarche n'est bien sûr pas sans conséquences. L'une d'elle – très positive – consiste à écrire des Spécifications Techniques de Besoin souvent mieux dimensionnées par rapport aux besoins réels. Toutefois, certaines contraintes perdureront comme le besoin de pouvoir fonctionner dans des environnements difficiles. Il n'y a, en effet, aucune raison que les profils et théâtres d'opération des missions militaires changent. L'élévation de température peut aussi être due à un échauffement cinétique (exemple : un missile en vol libre). Même si on peut s'attendre à des progrès dans la gestion des calories, cela ne changera probablement pas radicalement le problème au niveau des composants électroniques.

Il faut noter l'impact que peut avoir la permanente diminution des lithographies, diminution qui peut engendrer d'autres phénomènes (SEU : Single Event Upset).

L'utilisation de concepts civils

Le sujet peut être abordé sous deux aspects.

Choix de standards / protocoles et d'éléments d'architecture de systèmes et de calculateurs civils : C'est le 1^{er} aspect. Pendant la tâche d'architecture, un concepteur peut ainsi sélectionner un ou plusieurs standard(s) (exemples : USB, IEEE 1394, PCI...). Il bénéficiera ainsi du support de la très large communauté utilisatrice : existence de la norme, des outils, des composants (virtuels et réels, attention au risque de pérennité pour ces derniers)... Même s'il décide de n'utiliser qu'une partie de ce dont il peut disposer, il sera largement gagnant en terme de temps (et donc de coût) de développement au moins. Ceci dit, il faut se prévenir de l'idée consistant à considérer que, parce que la norme existe et décrit un protocole, tout le monde – y compris les néophytes – pourront prendre en charge une conception. Les protocoles sont complexes et une simple lecture même approfondie d'un document outre qu'elle est franchement rébarbative est loin de remplacer l'expérience.

En tout état de cause, il s'agit d'une démarche extrêmement positive qu'il faut encourager. Le risque essentiel est de sélectionner un standard devenant obsolète rapidement, risque limité si un minimum de soin est apporté lors du choix.

Utiliser des composants issus du monde civil : Il s'agit du 2^{ième} aspect. La tâche n'est pas si aisée qu'il y paraît.

Le problème de la gamme de température : Il est nécessaire de prévoir la mise en œuvre de ces composants sur une gamme de température élargie. Accessible pour des composants simples (transistors) ou à structure régulière (mémoires), l'exercice se complique notamment pour des circuits complexes tels que des processeurs. Ces derniers peuvent, par exemple, comporter des structures en partie asynchrones visant à optimiser les performances mais qui ne sont validées par le fournisseur que sur la gamme de température spécifiée. En cas d'utilisation sur une gamme élargie, il y a alors des risques de conflits internes liés à des temps de propagation tangents (courses de chemin). On constate des comportements aléatoires sur une ou plusieurs plage(s) de température plus ou moins réduite(s).

Par ailleurs, l'idée consistant à dire que « nos besoins étant proches de ceux de l'automobile, nous aurons là une source d'approvisionnement nous convenant » pourrait bien de se révéler fausse. En effet, s'il est vrai que les contraintes sont similaires, il est plus que probable que l'industrie automobile va s'orienter vers la conception de SoC, donc de circuits dédiés inaptes à remplir nos fonctionnalités.

Le problème de la pérennité : Les cycles des composants utilisés pour des applications civils sont sans commune mesure avec les besoins des militaires. Il ne s'agit même

plus de risques mais d'un élément à considérer de base : Il faudra faire évoluer la définition de tout équipement militaire tout au long de sa durée de vie pour traiter les problèmes d'obsolescence. Le cas le plus simple est lorsqu'il suffit de remplacer un circuit par un autre de fonctionnalité équivalente. Exemple type : les mémoires, pour peu que la carte ait été conçue de manière à pouvoir câbler des circuits de capacité plus importante. L'autre situation extrême, beaucoup plus difficile, est lorsque qu'il n'est plus possible de trouver un composant équivalent. Dans ce cas, il faut au moins prévoir une reprise de la carte et des couches basses du logiciel.

Disponibilité des composants : Certains composants, essentiellement dédiés aux applications Télécom. ou Automobile par exemple, risquent de ne plus exister sous leur forme classique mais uniquement virtuelle.

Information des fournisseurs : Bien entendu, dans tous les cas de figure, les fournisseurs restent plutôt avares en information. Nous serons donc tenu au courant des disparitions de composants de manière parcellaire et quant à obtenir des données détaillées sur ce qu'il est nécessaire de tester et comment pour envisager d'utiliser des circuits sur une gamme de température étendue, là c'est du domaine du rêve. Non seulement, ils n'y ont aucun intérêt financier, mais en plus ils ne voudront sûrement pas s'engager à nous fournir des informations et, en plus, à les tenir à jour en cas d'évolutions.

UNE SOLUTION ALTERNATIVE : LE « SYSTEM ON CHIP »

Devant un tableau, il faut bien le dire, un peu noir, comment pouvons nous réagir. Il y a probablement plusieurs possibilités, mais dans ce papier, nous nous contenterons d'en aborder une : Les « Systems On Chip » ou « SoC ».

Définition

Depuis déjà plusieurs années, les progrès des lithographies sont impressionnantes et permettent d'intégrer dans un seul circuit plusieurs millions de portes. Cela a permis de développer des processeurs puissants, mais ceux ci ne représentent finalement « que » le marché des PC et des stations de travail. Il y a bien d'autres applications industrielles ou grand public qui peuvent bénéficier de ces possibilités et sont à l'origine même du concept de System On Chip.

Un SoC est un circuit dédié, intégrant sinon toutes, du moins les principales fonctions d'un calculateur. Elles sont relatives à chaque application, mais on retrouve typiquement :

- 1 ou plusieurs cœur(s) de processeur
- 1 ou plusieurs cœur(s) de DSP (Digital Signal Processing)

- des périphériques :
- ✓ des interfaces bus système (ARINC, MIL-STD-1553...)
- ✓ des gestionnaires de liaisons séries
- ✓ des ports parallèles
- ✓ des timers / horloges temps réel
- ✓ des contrôleurs de commande moteur (générateurs PWM)
- et bien entendu – ce serait dommage de ne pas en profiter – des blocs de logique dédiée.

La figure 1 propose un synoptique générique d'un SoC. On retrouve finalement des notions proches de celle

d'une structure de calculateur classique, avec des blocs fonctionnels connectés à un bus on-chip. Pour ce dernier, il n'est, malgré tout, pas facilement imaginable de reprendre des standards classiques tel que, par exemple un bus PCI. En effet, certaines contraintes liées à la technologie sont à considérer (exemple : éviter d'avoir des potentiels flottants en interne circuit, donc par de lignes 3 états...). Toutefois, des standards apparaissent. Il convient de noter les efforts sur ce sujet d'organismes tel que VSIA (Virtual Chip Interface Alliance).

Il est aussi possible de prévoir des blocs analogiques ainsi que des convertisseurs analogiques / numériques et inversement.

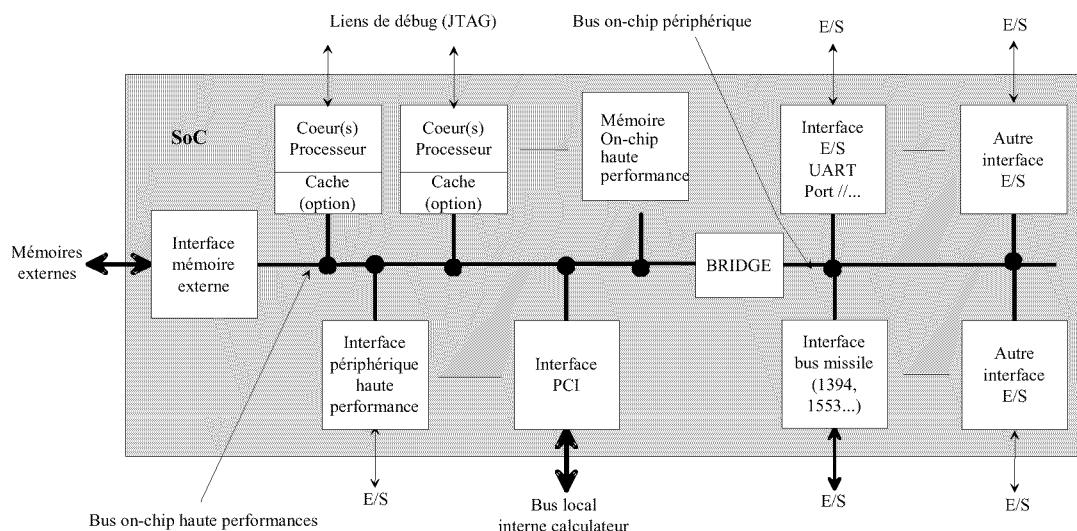


Figure 1 : Synoptique générique d'un SoC

Les marchés

Le principal marché à l'origine de cette tendance est indubitablement celui des télécommunications qui allie à la fois :

- un fort besoin d'intégration pour répondre aux attentes des consommateurs quant au poids et à l'autonomie des téléphones portables
- mais aussi de gros volumes, ce dont se félicitent les fondeurs.

D'autres applications apparaissent, tels que les équipements audio / vidéo et l'automobile.

L'ensemble des marchés cités est caractérisé par de gros volumes en production associés à des coûts très faibles.

Ce qui est aussi certain, c'est qu'on ne voit aucun signe laissant penser à une inversion de tendance. C'est plutôt le contraire, il est probable que l'on verra apparaître de nouveaux débouchés dans les domaines industriels et surtout grand public.

Les grands fournisseurs (d'outils entre autres mais pas exclusivement) l'on bien compris : Il suffit de faire un

passage sur les sites web respectifs ou d'assister à quelques conférences pour en être convaincu. Tout tourne autour du SoC, à un point tel qu'il vaut mieux être un peu méfiant vis à vis de ce qui s'en réclame...

Il en est de même au niveau de la presse : Il n'est pas une seule publication qui n'aît pas son lot de références au SoC !

Il ne s'agit donc pas simplement d'un effet de mode, mais d'une tendance bien réelle et durable.

Quelques définitions complémentaires

Avant d'aller plus loin, il est nécessaire de préciser certaines définitions.

Les circuits dédiés. Par circuits dédiés, on entend ASIC (Application Specific Integrated Circuit) ou FPGA (Field Programmable Gate Array).

Le synoptique ci dessous (figure 2) résume les principales étapes d'un développement type pour ces circuits. Alors qu'il apparaît linéaire, il ne l'est pas dans la réalité. Par exemple, il est certain qu'il y a un

rebourrage entre les phases « Faisabilité », « STB » et « Architecture ». De même, si un problème est découvert durant une phase de vérification, il y a correction, celle ci devant être effectuée généralement dans l'étape précédente ou encore avant. Par exemple, un problème découvert en Vérification « 2 » peut devoir être corrigé en retournant à l'étape de « Modélisation ». Enfin certaines tâches n'ont pas été mentionnées pour ne pas surcharger le synoptique. Elles n'en sont pas moins importantes. Il s'agit de l'insertion de dispositifs de test ou encore du floorplan (ou pré-placement, indispensable pour préparer le routage et éviter des problèmes lors de l'étape de Vérification « 3 »).

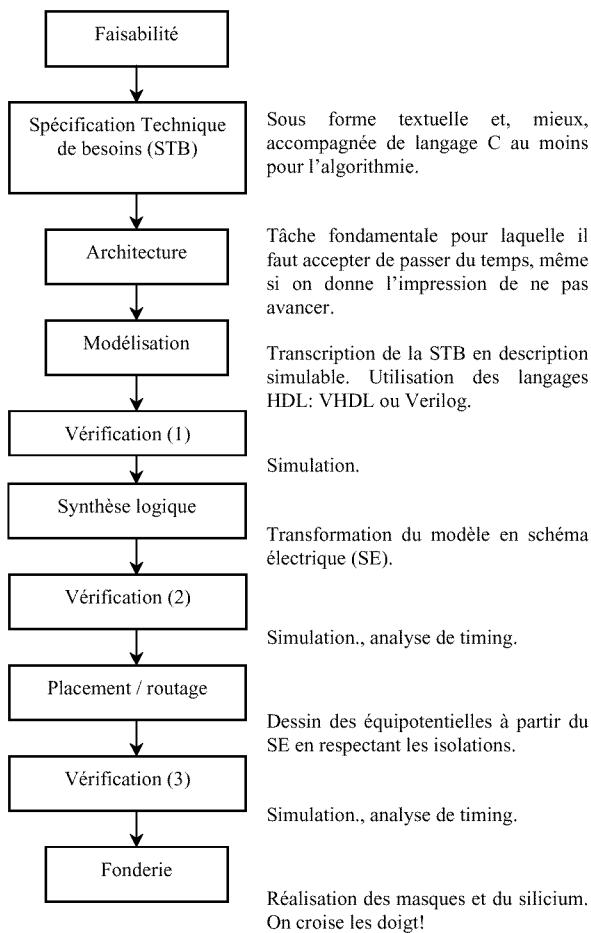


Figure 2 : Flot de conception d'un circuit personnalisé

Les flots ASIC et FPGA restent très voisins, étant entendu qu'il n'y a évidemment pas de fonderie dans le cas des FPGA mais la programmation d'une mémoire.

Précisons que la portée de ce § est bien limitée aux circuits personnalisés, ce qui ne correspond qu'à une phase du développement d'un SoC, dont le flot complet sera traité après

Les IP ou « Intellectual Property ». La notion de SoC est indissociable de celle d'IP (sans pour autant en avoir

l'exclusivité). Les IP peuvent être définies comme des composants virtuels. Il s'agit donc de blocs fonctionnels qui peuvent être achetés. Le marché est particulièrement actif et l'offre très fournie. On trouve la plupart des fonctions listées au § « Définition du SoC ». Il est aussi possible de les développer en interne société. Elles correspondent alors à des fonctions très spécifiques à l'activité et que l'on souhaite pouvoir réutiliser dans plusieurs applications.

Il est intéressant de rentrer un peu plus dans le détail, sachant que la maîtrise sur le long terme d'un SoC (ou de n'importe quel circuit mettant en œuvre un IP) est fonction du type d'IP utilisé (ou instancié). On distingue :

- *Les Soft IP* : Les blocs fonctionnels sont décrits en langage HDL et sont accompagnés de testbenches et de directives de synthèse logique. Celle solution permet de maîtriser entièrement le modèle du SoC et assure une bonne pérennité à long terme. Par contre, elle demande un peu plus de travail lors de la phase de développement.
- *Les Firm IP* : Les blocs fonctionnels sont fournis sous forme de schémas électriques. Là encore, des testbenches sont disponibles ainsi qu'un modèle comportemental autorisant les simulations de haut niveau (Vérification « 1 »). Dans ce cas, l'effort de conception est, bien entendu, moindre par rapport au cas précédent. Par contre on ne maîtrise pas du tous les aspects fonctionnels. Cela peut être très pénible durant le développement si le bloc n'est pas correctement développé et validé. De plus, les modifications éventuelles ultérieures du SoC seront plus délicates.
- *Les Hard IP* : Le bloc fonctionnel est, dans ce cas, synthétisé, placé et routé par le fondeur. Pour certains blocs, critiques en terme de performances en vitesse, ce choix est le meilleur. Toutefois, il est clair que l'on est entièrement dépendant de la politique du fournisseur : On n'a aucune maîtrise sur les aspects fonctionnels ni, pire encore, sur la pérennité. Le fondeur peut très bien décider de ne pas porter une Hard IP vers une nouvelle technologie tout en arrêtant celle utilisée. La situation alors délicate à gérer, encore plus s'il s'agit d'un cœur CPU, avec les impacts logiciels associés...

Les conditions d'accès sont très variées, rien de bien stabilisé n'ayant déjà été instauré. En général, il est nécessaire d'acquitter un coût d'accès à la licence, avec en plus des royalties sur les circuits en production. Notons que, dans certains cas, il n'y a pas de royalties et qu'il est aussi possible de disposer d'IP sans droit d'accès à la licence (même pour certains IP considérés comme complexes tel que des coeurs CPU) ! Ceci dit, le budget IP est généralement important, et dépendant de leur type. Les Soft IP sont, bien sûr, les plus chers.

Outre le classement précédent, on distingue généralement 2 grandes catégories :

- *Les « Commodity IP »* : Ce sont les blocs d'usage courant. On retrouve les fonctions PCI, USB, UART... L'offre est très riche.
- *Les « Stars IP »* : On y classe traditionnellement les cœurs de processeur et les fonctions émergentes. Toutefois, force est de constater que l'on assiste, en particulier pour les cœurs de processeur, à des situations un peu abusives. La complexité d'un tel cœur est, approximativement, de 40 000 portes (RISC 32 bits sans opérateurs flottants), ce qui n'est pas énorme. Les coûts (notamment des soft IP), par contre, sont généralement très élevés.

Lors de la sélection des IP, il importe de prendre en considération :

- *Les coûts* : Accès licence, royalties mais aussi support / maintenance.
- *La qualité des fournitures*. Il importe de savoir si un minimum de règles de conception des IP a bien été respecté : Conception synchrone, sur fronts montants

uniquement, ... La facilité de l'instanciation des blocs dans le modèle (et donc coûts et délais associés) en dépend largement. Les publications à ce sujet sont nombreuses, on citera, par exemple, l'initiative OpenMore de Mentor / Synopsys.

Actuellement, la tendance pour les « Stars IP » est très orientée vers les Hard IP (pour des raisons de coûts !) et vers les Soft IP pour les autres. Notons toutefois une évolution encore assez récente qui apparaît, celle des « Soft IP configurables ». Il s'agit de générer un bloc suivant les besoins spécifiques (dans une certaine mesure) de l'utilisateur. Ainsi, ARC propose, moyennant le chargement via Internet, d'un programme permettant de créer un cœur de DSP et l'environnement de développement logiciel en fonction d'un certain nombre de choix. Tensilica offre une approche similaire peut être plus aboutie, aussi pour un cœur processeur : Les besoins utilisateurs sont décrits via Internet et la Soft IP est générée par Tensilica et transférée, toujours via le net. Cette voie permet des options de configuration plus complexes. Nul doute que cette voie a de l'avenir, ainsi qu'Internet comme média de communication et d'échange.

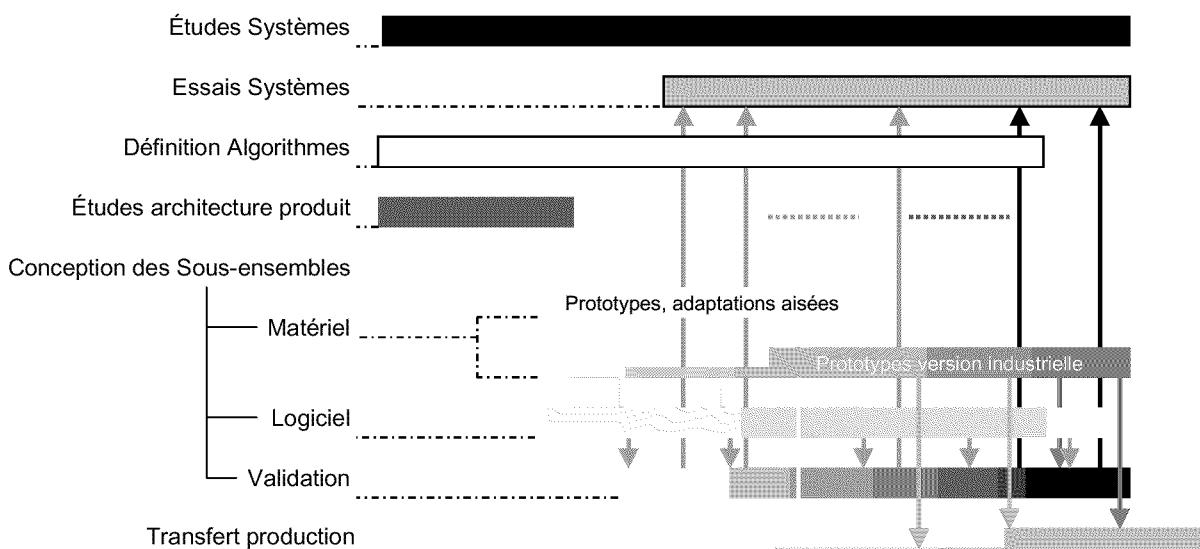


Figure 3 : Flot de développement d'un SoC

FLOT DE DEVELOPPEMENT D'UN SoC

Là encore, nombre de publications existent sur le sujet. Une chose est sûre, il n'y a pas de flot standard, applicable quel que soit le type d'activité de la société. Dans ce §, on va donc s'attacher à préciser les grandes étapes à prévoir avec les différentes options possibles pour chacune d'elles ainsi que quelques écueils à éviter.

On partira pour ce faire du synoptique de la figure 3.

Le flot proposé est basé sur la réalisation d'une maquette à base de FPGA en plus du développement des

prototypes en forme mettant en œuvre le SoC. C'est un choix dont l'intérêt est décrit dans un des § suivants.

Implication des équipes Système

Le 1^{er} point à préciser est qu'il n'est pas imaginable de dissocier les aspects Système / Algorithmique du processus de conception d'un SoC. Si cette démarche est sans doute très présente dans la culture d'entreprises du domaine des Télécommunications, elle l'est beaucoup moins dans des sociétés, par exemple de l'aéronautique. Deux raisons à cela :

- L'historique des Sociétés
- Les différences existant au niveau de la notion de système. Dans le domaine de l'aéronautique, le système met en œuvre un ensemble d'équipements complexes, pas nécessairement basés sur les mêmes technologies (mécaniques, optiques...).

Les responsables du Système et de la définition des algorithmes doivent être très impliqués dans les études d'architecture. Il est nécessaire d'optimiser les algorithmes et de les orienter en vue d'en faciliter l'implémentation. Cette démarche était classique, voire naturelle au tout début de l'électronique essentiellement à cause des limitations de cette technologie naissante. Elle s'est - à tort - largement affaiblie avec la généralisation des structures programmables. Il paraît sain de la remettre au goût du jour non pas du fait de limitations technologiques mais plutôt de coûts en production.

Par ailleurs, en cas d'erreur découverte une fois le SoC réalisé, toute modification risque d'être longue et coûteuse si une solution logiciel n'est pas applicable. Fort heureusement, on commence à voir apparaître des outils permettant la fourniture de Spécifications Techniques de Besoins (STB) simulables donc d'une part validées par rapport aux besoins et d'autre part pouvant servir de modèle de référence pour la conception du SoC. Il convient enfin de noter qu'il est beaucoup plus difficile de valider par simulation des événements asynchrones que synchrones. Même si ce n'est pas toujours possible, on cherchera à les éviter.

Le plan de développement

Le 2^{ième} point important est de définir en début d'affaire le plan de développement et de préciser entre autres :

- S'il est nécessaire de passer par une phase maquette.
- Les différentes étapes de validation et les entrées nécessaires à celles ci.
- Le planning, bien entendu, avec le calage du début du développement du prototype en forme.

Passage par une phase maquette

Il est fortement souhaitable pour :

- Pouvoir mettre au plus vite à disposition des équipes Système des versions intermédiaires de calculateur.
- Envisager l'intégration matériel / logiciel avant de pouvoir disposer du prototype en forme.

Besoin des équipes Système. Disposer au plus vite de calculateurs permet aux équipes Système de commencer progressivement les intégrations. Il est clair qu'au début de celles-ci, il n'est pas nécessaire de pouvoir activer toutes les fonctionnalités prévues pour le calculateur. La gestion des E/S sera donc initialement proposée et complétée progressivement en fonction des souhaits émis au niveau Système.

Cette démarche est aussi une aide pour les concepteurs du calculateur. Même si les simulations permettent d'aller très loin dans la validation, on ne peut simuler que ce dont on a les modèles. Ce n'est pas nécessairement le cas de tous les éléments connectés au calculateur. Dans ce cas, on écrit un modèle, mais qui, souvent simplifié, ne représente pas toujours le comportement réel. Les essais avec les équipements réels sont de bon tests.

Intégration Logiciel. L'autre intérêt majeur de disposer d'une maquette est de pouvoir commencer l'intégration Logiciel avant d'avoir le prototype en forme. A ce sujet, on voit apparaître une multitude d'émulateurs ou d'accélérateurs Matériel dont on retrouvera l'intérêt au niveau des simulations après synthèse. Ce genre de moyen peut être utilisé pour faciliter l'intégration logiciel lorsque l'on cherche à effectuer celle ci en utilisant le modèle HDL du SoC. Complété avec des environnements de co-simulation (tel que Seamless de Mentor ou Eagle-i de Synopsys...), il est possible, pour les équipes logiciel et de développement du SoC de travailler chacune dans leur environnement. De plus, en simulation, tous les nœuds internes du circuit sont accessibles, ce qui facilite grandement le débogage. Enfin, au niveau du SoC, le logiciel applicatif constitue un testbench rêvé. Malheureusement, il convient de rester réaliste : La puissance des machines, même associées à un accélérateur, reste en deçà des besoins nécessaires à la simulation d'un logiciel applicatif complet. On limitera donc cette approche au niveau des handlers de base du logiciel.

La mise en œuvre d'accélérateurs n'est pas évidente. Ils représentent un investissement conséquent et reposent soit sur des structures propriétaires soit, c'est de plus en plus souvent le cas, sur des FPGA. Dans ce cas, bien souvent, il faut aussi disposer de la chaîne de développement FPGA. Il sera nécessaire de considérer, pour le développement, la tâche de partitionnement vers plusieurs FPGA avec aussi les étapes de placement / routage de chacun d'eux. C'est assez lourd... et redondant avec la maquette. Enfin dernier inconvénient relatif à ce genre d'investissement : La pérennité est limitée ! En effet, ces moyens reposent sur des technologies très évolutives (FPGA), donc rapidement obsolètes.

Les étapes de validation

Le synoptique de la figure 3 le montre, les étapes de validation sont nombreuses et menées à bien par des équipes différentes. Par ailleurs, on s'en doute bien, elles sont fondamentales, d'où l'intérêt de les préparer soigneusement. Il est donc largement souhaitable d'établir un plan de validation le plus tôt possible après le plan de développement et décrivant chaque étape de validation avec les entrées / sorties, qui fournit quoi et qui fait quoi. Cela présente les avantages :

- de s'assurer que rien n'aura été oublié (validation de certains modes fonctionnels)

- de ne pas non plus valider 2 fois la même chose, ce qui peut coûter cher en temps
- de fiabiliser le planning de développement, en précisant les responsabilités.

Ce plan de validation doit être tenu à jour et enrichit en permanence.

On se limitera, dans ce §, aux validations fonctionnelles. Les techniques de validation sont essentiellement basées sur des simulations, associées à des essais effectués avec les maquettes et prototypes.

Les simulations. Plus les simulations sont faites à haut niveau, plus elles sont rapides. On a donc tout intérêt à envisager de commencer au niveau Système, en créant des Spécifications Techniques de Besoins simulables. Les validations de haut niveau peuvent ainsi être faites directement par l'équipe système ce qui est plus efficace.

Au niveau des modèles développés pour les besoins de la phase Maquettage, la priorité est d'en disposer rapidement. Les simulations sont donc réduites (ce qui ne veut pas dire supprimées), mais complétées par des essais sur table.

Le SoC, quant à lui, doit être validé de manière intensive et à toutes les étapes de conception. Il convient de noter que les simulations fonctionnelles sont longues.

Au niveau du modèle HDL, le temps passé est essentiellement dû à la définition des thèmes de test et à la création des testbenches correspondants. De plus, il est souvent difficile de savoir si la validation est exhaustive ou non. C'est particulièrement vrai pour des applications de traitement d'images. Par contre, les temps machine restent acceptables.

Par ailleurs, il faudra bien comparer les résultats de simulation par rapport à une référence et ce de manière automatique. On voit bien, là encore, l'intérêt de disposer d'une Spécification Technique de Besoins simulable.

Pour revenir à la définition des thèmes de test, il est important d'y associer, bien sûr les concepteurs du SoC, mais aussi les utilisateurs :

- au niveau système, y compris les responsables de la définition des algorithmes, et ce pour éviter toute incompréhension
- au niveau carte, de manière à fiabiliser la définition des interfaces avec le reste des circuits des cartes.
- au niveau logiciel, c'est aussi à ce niveau que l'utilisation des handlers logiciel de base constitue un excellent testbench, ainsi que cela a déjà été précisé.

Après synthèse et placement / routage, les temps des simulations fonctionnelles sont fondamentalement prohibitifs. Pour assurer la cohérence des tests, elles ont pour base les testbenches définis au niveau de la validation du modèle HDL. Mais, du fait de la forte complexité de ces circuits, les temps machine sont importants, pouvant durer plusieurs semaines (sur

plusieurs stations de travail). On limite donc ce type de simulation au strict minimum et on s'attache plutôt à utiliser un outil d'analyse statique de timing. Celui ci est, de toute façon indispensable à une validation correcte au niveau structurel (après synthèse). Eventuellement, des outils de preuve formelle peuvent constituer aussi une aide mais ils ne marchent généralement pas très bien pour des structures algorithmiques complexes, du moins actuellement.

Essais sur table. Ce type d'essai est très classique. On notera tout de même :

- l'intérêt qu'il y a de bien les formaliser de manière à éventuellement les transformer en testbench
- en cas de problème, l'importance qu'il y a de fiabiliser la cohérence des prises en compte des corrections au niveau des maquettes (FPGA) mais aussi au niveau du SoC

Enfin, même sur maquette, en cas de problème, il est souvent plus facile d'en trouver l'origine en simulation où l'on a accès à l'ensemble de noeud du circuit, plutôt que sur la réalisation matérielle.

La gestion des plannings

On l'a vu, une bonne gestion des étapes critiques permet logiquement de fiabiliser le planning de développement. L'autre aspect à maîtriser est la relation avec les intervenants extérieurs.

Sous-traitance de conception : Elle n'est, bien sûr, pas obligatoire. Dans le cas où on y a recourt, en particulier pour la conception de blocs fonctionnels en HDL, il est souhaitable d'imposer des règles de conception qui permettent de s'assurer de la qualité de conception et d'un interfaçage aisément avec le reste du circuit

Là encore, les recommandations définies dans le cadre de l'initiative OpenMore sont une excellente base.

Il est aussi largement préférable que le sous-traitant utilise les mêmes outils de conception que ceux mis en œuvre dans la société.

Placement / routage, relations avec le fondeur : Il s'agit d'un cas un peu particulier de sous-traitance. C'est souvent le fondeur qui se charge du placement / routage étape souvent assez délicate et qui, en cas de problème peut durer fort longtemps. Deux règles essentielles sont à respecter :

- Même si l'on cherche à être indépendant des technologies, il est souhaitable de sélectionner le fondeur au plus tôt, et de mettre en place des échanges sur l'avancement du projet et les règles de conception qu'il peut vouloir imposer.
- Avant d'envoyer un circuit pour le placement / routage, on aura tout intérêt à passer par une étape de pré-placement, et de prendre en

compte le routage de(s) horloge(s). Cela évite nombre d'aller / retours, souvent pénibles et longs.

Plannings trop optimisés : Bien sûr, le développement doit être le plus court possible. Il ne faut tout de même pas tomber dans le piège consistant à «oublier» des tâches. En particulier, il ne faut pas considérer que, parce qu'on utilise des IP, il n'y a aucun effort de conception à prévoir. Il y a généralement une logique d'interfaçage à concevoir. Un bon exemple est celui du couplage d'un cœur CPU sur un bus on-chip. Il serait bien surprenant que les 2 blocs aient été conçus en même temps, sans évolution aucune de l'un par rapport à l'autre.

Recouvrement des activités de conception FPGA (maquette) / SoC : A condition de bien le prévoir au niveau de l'architecture des circuits, il est imaginable de reprendre des blocs HDL développé pour la maquette pour le SoC proprement dit et inversement. Par contre, il faudra faire attention à la cohérence des fichiers, en particulier, suite à la prise en compte des évolutions.

Intégration Matériel / Logiciel et Système

Ce thème a déjà été abordé pour justifier l'intérêt de la phase Maquette. Toutefois, à un moment donné, il faudra bien intégrer le prototype en forme avec le SoC... et bien sûr, il y aura des problèmes !

Au niveau logiciel, les techniques de débogage n'ont aucune raison d'être différentes de celles utilisées avec des structures programmables classiques.

Au niveau système, il est certain qu'il faut prévoir des dispositifs intégrés dans le SoC et permettant – au minimum – de connaître les données échangées entre blocs.

Enfin, rappelons l'intérêt qu'il y a à disposer d'un modèle et permettant un débogage souvent facilité, moyennant la définition d'un test précis.

TECHNOLOGIE ET PERENNITE

Avantages / inconvénients de l'approche SoC

Les avantages de l'approche décrite se retrouve à divers niveaux :

Les aspects thermiques, la consommation : L'intégration silicium permet souvent de diminuer la puissance dissipée. Si la consommation n'est généralement pas un problème pour des applications aéronautiques, les aspects thermiques eux le sont. Le choix d'une technologie ASIC (avec fonderie) par rapport à une filière FPGA est, à ce sujet préférable.

Les coûts en production : Les structures de ce type sont beaucoup plus simples, mettant en œuvre moins de cartes et moins de composants. Autant de facteurs réducteurs de

coût en production. La consommation diminuant, on gagne aussi sur la complexité des blocs alimentation.

Les dispositifs de drainage des calories se simplifient.

La fiabilité : La maîtrise de la consommation et la réduction du nombre de composants permettent d'améliorer la fiabilité des calculateurs.

Les inconvénients : Bien sûr, il y en a toujours :

Accès fonderie : Dans le cas du développement d'un SoC basé sur une technologie ASIC (par opposition à FPGA). La lithographie à utiliser sera du 0.25 ou 0.18 µm. Les coûts de réalisation des masques sont importants. De plus, les fondeurs, surtout actuellement, recherchent de gros volumes ce qui n'est absolument pas notre cas au niveau des applications aéronautiques. Il convient de remarquer, pour ces 2 aspects que ce débat sur l'accès aux fonderies date du début de la technologie ASIC. Jusqu'à maintenant, nous avons toujours trouvé des fondeurs acceptant de travailler avec nous. Par ailleurs, des techniques d'accès multi-projects wafer et multi level mask se développent, limitant l'envolé du prix des masques.

Conditions d'achat : C'est bien connu : Les sociétés ont horreur d'acheter de grosses quantités couvrant les besoins de plusieurs années. C'est malgré tout ce qu'il faut se préparer à faire ! En effet, pour un SoC développé en technologie ASIC, on demandera au fondeur de lancer un batch de wafers, ce qui représentera sans doute si ce n'est toute, au moins une bonne partie de la série ! Ceci dit, ce problème est aussi de plus en plus souvent rencontré dans le cadre de l'utilisation de composants civils, et donc de FPGA.

Risques et coûts de développement : Dès que l'on parle ASIC, on voit souvent les cheveux de dresser sur la tête de nos interlocuteurs ! A tord ! Finalement, le coût de développement d'une fonction en ASIC est similaire à celui d'une structure programmable aux coûts de fonderie près. Même si ces derniers sont élevés, ils ne sont malgré tout pas rédhibitoires par rapport aux coûts de développement d'un calculateur. Par ailleurs, les outils et les méthodologies utilisés permettent de largement fiabiliser le développement et de donc de réduire les risques.

Souplesse : Bien sûr, une fois la phase de fonderie terminée, il est sinon difficile en tout cas long et coûteux de modifier une fonction pour prendre en compte une évolution. Il est important d'insister là sur l'étape d'architecture de manière à rendre la structure paramétrable (exemple : pouvoir modifier les paramètres d'un filtre par programmation). De plus, on l'a vu, le SoC intègre beaucoup de blocs standards, le tout contrôlé par logiciel du fait de l'intégration d'un cœur CPU (ou plusieurs). L'idée de circuits ASIC rigides n'est donc plus vraiment de mise et, en tout cas ne l'a jamais été pour les technologies FPGA. Enfin, le côté positif est que « Réfléchir avant d'agir » permet souvent d'éviter des erreurs et de gagner en temps de développement...

SoC : Le choix FPGA / ASIC

Même si la décision est prise de s'orienter vers du FPGA, il est souhaitable de conserver la possibilité de migrer vers de l'ASIC et donc d'appliquer les règles idoines. On commence à voir apparaître des FPGA intégrant des coeurs de processeur : Attention à la compatibilité avec une possible filière ASIC.

Le FPGA est, bien entendu, plus souple en développement en cas d'évolution mais est notamment plus cher en production. Il faut donc faire un bilan économique en fonction des quantités à produire.

Le FPGA présente aussi le risque d'obsolescence. Alors que les mémoires SRAM étaient le véhicule technologique utilisé par les fondeurs pour mettre au point les nouvelles lithographies, elles ont été remplacées par les FPGA qui ont aussi une structure régulière et pour lesquels on assiste à l'envolé des complexités. Ces composants sont donc à la pointe des technologies, mais cette course risque de laisser rapidement de côté les versions à peine plus anciennes. Il y aura, bien sûr, des FPGA permettant de remplir la fonction, mais la compatibilité avec la carte n'est pas assurée.

Au moment du choix, il convient d'être prudent par rapport à la réalité des complexités annoncées par les fournisseurs de FPGA. 1 million de portes FPGA est loin d'être équivalent à 1 million de parties ASIC ! En plus le rapport est variable en fonction des fournisseurs et même des familles. Idem pour les vitesses qui correspondent souvent à des « best case », c'est à dire la fréquence maximum avec des fan out de 1... ce qui est rarement le cas dans la réalité !

La pérennité

Ce thème est la trame de ce document en entier, avec, toutefois, quelques aspects complémentaires.

Propriété du circuit : La société est propriétaire de son design et du modèle HDL. Moyennant le respect de quelques règles, le portage d'une technologie vers une autre n'est pas un problème majeur.

Une limitation existe en cas d'utilisation de Firm ou Hard IP, pour lesquelles il faudra se limiter à de grands standards du marché.

A noter, que, à ce niveau, il ne s'agit pas de dire que la technologie (lithographie) ne deviendra pas obsolète. Par contre le fait de pouvoir migrer vers une autre, même si les coûts ne sont pas nuls, permet d'éviter la reprise de cartes et pire encore du logiciel.

Conditions d'achat : Celles là même évoquées précédemment et présentées comme un inconvénient ! Le fait de faire du stock en quantité suffisante pour la série met à l'abri de tout surcoût du fait de problèmes d'obsolescence...

Gamme de température de fonctionnement : A partir du moment où la société contrôle l'ensemble du développement, il est parfaitement possible de valider le SoC (les timings) sur une gamme de température correspondant à nos besoins. En production, si le fondeur refuse d'effectuer les tests en température, il est encore possible de réaliser ou de faire réaliser des tests spécifiques puisque l'on est aussi propriétaire des vecteurs de test.

CONCLUSION

L'approche SoC nécessite d'adapter certaines méthodes de développement au niveau validation par exemple et d'effectuer un contrôle rigoureux du développement. Par contre, elle est particulièrement attractive sur les plans des performances au sens large mais aussi de la pérennité.

This page has been deliberately left blank

Page intentionnellement blanche